Lab 3: Symmetric key cryptography

Zadatak: dešifrirati ciphertext korištenjem **brute-force** napada.

Preuzeli smo datoteke koje su naši potencijalni izazovi. Trebali smo hashirati svoje ime i prezime kako bi usporedbom te vrijednosti i naziva datoteka saznali koji izazov pripada nama.

Kod koji smo za to koristili:

```
from cryptography.hazmat.primitives import hashes

def hash(input):
    if not isinstance(input, bytes):
        input = input.encode()

    digest = hashes.Hash(hashes.SHA256())
    digest.update(input)
    hash = digest.finalize()

    return hash.hex()

filename = hash('jerkovic_ela') + ".encrypted
```

Challenge koji smo dobili je enkriptiran **simetričnim ključem** pa nam za dekripciju treba isti taj ključ. Entropija ključa je **22 bita** tj. svi bitovi su 0 osim zadnjih 22 koji su generirani slučajno.

Kako bi riješili challenge trebali smo koristiti **brute-force** napad (pretpostaviti neki ključ, dekriptirati tim ključem i vidjeti ima li rješenje smisla)

Pretpostavili smo da je enkriptirana datoteka slika u png formatu pa smo zaključili da plaintext započinje sa 8 byteova karakterističnih za png format. Tako znamo i plaintext i cyphertext.

Dekripciju smo izvršavali pomoću funkcije iz Fernet libarya.

Cijeli kod:

```
import base64
from os import path
from pydoc import plain
from cryptography.hazmat.primitives import hashes
from cryptography.fernet import Fernet

def hash(input):
    if not isinstance(input, bytes):
        input = input.encode()

    digest = hashes.Hash(hashes.SHA256())
    digest.update(input)
    hash = digest.finalize()

    return hash.hex()

def test_png(header):
    if header.startswith(b"\211PNG\r\n\032\n"):
        return True
    return False

def brute_force(ciphertext):
    ctr = 0
    while True:
        key_bytes = ctr.to_bytes(32, "big")
        key = base64.urlsafe_b64encode(key_bytes)

        # Now initialize the Fernet system with the given key
        # and try to decrypt your challenge.
        # Think, how do you know that the key tested is the correct key
        # (i.e., how do you break out of this infinite loop)?

        try:
            plaintext = Fernet(key).decrypt(ciphertext)
            header = plaintext[:32]

            if test_png(header):
                print(f"BINGO:{key}")
                with open("BINGO.png", "wb") as file:
                    file.write(plaintext)
                break
        except Exception:
            pass

        ctr += 1
        if not ctr % 1000:
            print(f"[*] Keys tested: {ctr:,}", end="\r")

if __name__ == "__main__":
    filename = hash('jerkovic_ela') + ".encrypted"
```

```
# Create a file with the filename if it does not already exist
if not path.exists(filename):
    with open(filename, "wb") as file:
        file.write(b"")

#Open your challenge file and read in your challenge
with open(filename, "rb") as file:
    ciphertext = file.read()

#print(ciphertext)

#Start the attack
brute_force(ciphertext)
```

Rezultat: